Towards minimum perceptual error training for DNN-based speech synthesis

Towards minimum perceptual error training for DNN-based speech synthesis

Cassia Valentini-Botinhao, Zhizheng Wu, Simon King

The Centre for Speech Technology Research, University of Edinburgh, United Kingdom

cvbotinh@inf.ed.ac.uk {zhizheng.wu, simon.king}@ed.ac.uk

Abstract

We propose to use a perceptually-oriented domain to improve the quality of text-to-speech generated by deep neural networks (DNNs). We train a DNN that predicts the parameters required for speech reconstruction but whose cost function is calculated in another domain. In this paper, to represent this perceptual domain we extract an approximated version of the Spectro-Temporal Excitation Pattern that was originally proposed as part of a model of hearing speech in noise. We train DNNs that predict band aperiodicity, fundamental frequency and Mel cepstral coefficients and compare generated speech when the spectral cost function is defined in the Mel cepstral, warped log spectrum or perceptual domains. Objective results indicate that the perceptual domain system achieves the highest quality.

Index Terms: speech synthesis, DNN, objective measures, spectral analysis

1. Introduction

The quality of speech generated by statistical parametric systems has benefited from advances in acoustic models [1–6], vocoders [7, 8] and postfilters [9–11]. However the challenge of how to create truly high quality speech from learned vocoder parameters still remains. The vocoder itself is certainly one of the main limitations. But modelling assumptions, such as independence among different acoustic parameters, e.g., source and the filter, have also been shown to cause great degradation [12]. It is inevitable that any vocoder or statistic model will introduce error, so perhaps we should aim for errors that are introduced at any stage of the process to be as imperceptible as possible.

The idea of minimising a perceptual error is not new. Minimum Generation Error (MGE) [3, 13] for hidden Markov model (HMM)-based speech synthesis could be thought of as a step in this direction. In MGE training, the model parameters are updated not to maximize the likelihood of the data but to minimize the error between generated trajectories and ones extracted from natural speech. The error could be the Euclidean distance between generated and extracted trajectories [3] or a distance measured in a transformed domain like the log magnitude spectrum [13]. Unified feature extraction and model training could also lend itself to perceptual error minimisation [14, 15]. Nakamura et. al [14] proposed to extract Mel cepstral coefficients that maximize the likelihood of the data. More recently Shinji et. al introduced a compact representation of the spectrum using autoencoders [15]. Both techniques could be seen as error-minimising alternatives to Mel cepstral analysis [16].

The recent success of Deep Neural Network (DNN) speech synthesis [5, 6, 17, 18] suggest a range of new directions for minimum perceptual error training. In general, when training a DNN to predict acoustic parameters, all parameters are optimised using a shared cost function, allowing the model potentially to learn dependencies between output parameters.

DNN training easily allows for different cost functions to be used. It is possible to train a DNN to predict Mel cepstral coefficients but to calculate the error in the higher-dimensional spectral domain, simply by reformulating the cost function. It is also possible to train a DNN to predict the spectrum directly.

There are, however, more perceptually relevant representations of speech that could be used to measure the error, but that do not allow for synthesis. So, we might measure the error not directly on the output acoustic features (i.e., vocoder parameters) but in some other domain, which may not itself be useful for speech generation. In this situation, it is desirable to train a model that predicts vocoder parameters – necessary to eventually generate the waveform – but to calculate the error in this perceptual domain. In this paper we exploit this idea, using a particular perceptual representation of the speech spectrum.

Section 2 presents different spectral parametrisations followed by Section 3 where we propose minimum perceptual error training using such representations. We present our experimental design and results in Section 4 followed by discussions and conclusions.

2. Spectral parametrisation

We detail three spectral parametrisations: the Mel cepstral coefficients, the warped log magnitude spectrum and the Spectro-Temporal Excitation Pattern (STEP).

2.1. Mel cepstral coefficients

We can represent the spectrum $H(e^{j\omega})$ by a M-th order series of coefficients referred to as Mel cepstral coefficients $\{c_m\}_{m=0}^{M-1}$ following the relation:

$$H(e^{j\omega}) = \exp \sum_{m=0}^{M-1} c_m e^{-jm\,\tilde{\omega}} \tag{1}$$

$$\tilde{\omega} = \tan^{-1} \frac{(1-\alpha^2)\sin\omega}{(1+\alpha^2)\cos\omega - 2\alpha} \tag{2}$$

where α is a factor that controls warping in the frequency domain. We can choose α such that $\tilde{\omega}$ spans the frequency axis on a particular scale, for instance the Mel scale, leading to the so-called Mel cepstral coefficients [16].

The discrete log magnitude spectrum is defined by the Mel cepstral coefficients as follows:

$$\log|H(\omega_k)| = \sum_{m=0}^{M-1} c_m \cos(m\,\tilde{\omega}_k) \tag{3}$$

$$\mathbf{h} = \mathbf{D}_\alpha \mathbf{c} \tag{4}$$

where k=0 ... $N-1$ is an index that covers the frequency scale uniformly, $\mathbf{h}$ is a column vector of size N containing the log magnitude spectrum, $\mathbf{c}$ is the Mel cepstral coefficient column vector of size M and $\mathbf{D}_\alpha$ is an N by M matrix defined by α. This matrix corresponds to a warped discrete cosine transform. Note that $\mathbf{h}$ is in linear frequency.

If M is equal to N than it is possible to reverse the operation, i.e., to obtain the original log magnitude spectrum from the Mel cepstral coefficients. To represent the spectrum in a compact way M is usually set to be much smaller than N. To calculate M cepstral coefficients it is common to first extract N coefficients and than truncate the series. This method tends to overly smooth the spectrum. It is possible to extract a better fit of cepstral coefficients by using the Unbiased Estimator of the Log Spectrum (UELS) [19]. This method is often referred to as Mel cepstral analysis [16].

2.2. Warped log spectrum

The Mel cepstral representation is attractive due to its size and modelling power. The latter is mainly due to the frequency warping inspired by the frequency resolution of the cochlear membrane. Another potential perceptual domain is the warped log magnitude spectrum. It can be calculated from the Mel cepstral coefficients as follows:

$$\tilde{\mathbf{h}} = \mathbf{D}_0 \mathbf{c} \tag{5}$$

where $\tilde{\mathbf{h}}$ is an N size column vector containing the warped log magnitude spectrum. When $\mathbf{c}$ is of the same length as $\mathbf{h}$, i.e when M=N, this operation converts a linear log spectrum as described by $\mathbf{c}$ to a warped domain with no loss in information.

2.3. Spectro-Temporal Excitation Pattern

Finally we present a third alternative for spectral parametrisation. The Spectro-Temporal Excitation Pattern (STEP) representation was proposed in the context of the Glimpse model for speech perception in noise [20] and the Glimpse Proportion (GP) measure for intelligibility of speech in noise [21]. This measure is based on the idea that, in a noisy environment, humans focus their auditory attention on 'glimpses' of speech that are not masked by noise. To detect such glimpses, the STEP representations of speech and noise are compared. The GP measure correlates well with subjective scores for intelligibility of natural [21] and synthetic speech [22] in a variety of noises.

To represent a signal in terms of STEP we first decompose its waveform into different frequency channels using a Gammatone filterbank whose central frequencies are linearly spaced on the equivalent rectangular bandwidth scale [23]. For each channel, the temporal envelope is extracted with an absolute value operation, smoothed with a low pass filter and then averaged across limited time intervals. The temporal envelope and the low pass filtering elements are inspired by hair cell sensitivity and observed low frequency temporal modulation correlations with intelligibility. A glimpse is detected in a time frequency region when the speech STEP value in that region is larger than the noise value. In this paper, we assume that differences between the STEP representations of two speech signals will be a good measure of how different they sound.

The original GP measure is calculated directly from the waveform domain. The approximation proposed in [24] redefines it, so it can be computed from the magnitude spectrum whilst remaining well correlated with subjective intelligibility scores. The STEP representation at filter channel f is given by:

$$y_f = \log\left[\frac{1}{N}\left(\mathbf{G}_f\overline{\mathbf{h}} \circledN \mathbf{G}_f\overline{\mathbf{h}}\right)^\top \mathbf{S}\mathbf{b}\right] \tag{6}$$

where $\overline{\mathbf{h}} = \exp \mathbf{h}$ is the linear magnitude spectrum, $\mathbf{G}_f$ is an N by N diagonal matrix whose diagonal contains the Gammatone filter frequency response for channel f, $\mathbf{S}$ is an N by N diagonal matrix whose diagonal contains the frequency response of the smoothing filter, $\mathbf{b}$ a column vector of size N containing the coefficients of the average filter and $\circledN$ is a circular convolution operation of dimension N.

By exchanging the smoothing and the averaging operations, it is possible to represent the circular convolution using the real value discrete Fourier transform (DFT) matrix $\mathbf{F}$ as follows:

$$y_f = \log\left[\frac{s_f}{N^2}\ \left(((\mathbf{G}_f\overline{\mathbf{h}})^\top\mathbf{F}) \times ((\mathbf{G}_f\overline{\mathbf{h}})^\top\mathbf{F})\right)\mathbf{F}\ \mathbf{b}\right] \tag{7}$$

where $\mathbf{F}$ is the real part of the DFT matrix of size N by N, s_f is the smoothing factor applied to channel f and operator $\times$ is element-wise product.

3. Minimum perceptual error training

Parameter update using the back-propagation procedure to train a feed forward neural network is given by:

$$w_{i,j}^k = w_{i,j}^k - \rho\frac{\partial E}{\partial w_{i,j}^k} \tag{8}$$

$$\frac{\partial E}{\partial w_{i,j}^k} = \frac{\partial E}{\partial o_i^k}\frac{\partial o_i^k}{\partial w_{i,j}^k} \tag{9}$$

where $w_{i,j}^k$ is the weight between the i-th unit of the k-th layer and the j-th unit of the $(k-1)$-th layer, o_i^k is the output of unit i in layer k, E is the training error and ρ is the learning rate.

The definition of cost function E affects the propagation of the error through the layers, more specifically it changes the first term of the right hand side of the previous equation for the case where k is the output layer. For a linear output layer:

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial w_{i,j}} \tag{10}$$

where x_i is the output of the i-th unit of the output layer.

If the cost function is the sum of squared errors, the gradient with respect to the output layer $\mathbf{x}$ is:

$$E = \sum_{i=0}^{L}(x_i - x_i^*)^2 \qquad \frac{\partial E}{\partial \mathbf{x}} = 2(\mathbf{x} - \mathbf{x}^*) \tag{11}$$

where x_i^* is the i-th reference acoustic feature. In our case, $\mathbf{x}$ will be composed of Mel cepstral coefficients plus some acoustic parameters that describe the excitation (e.g., F0).

The gradient with respect to the Mel cepstral coefficients when the cost function is in the Mel cepstral domain is:

$$\frac{\partial E}{\partial \mathbf{c}} = 2(\mathbf{c} - \mathbf{c}^*) \tag{12}$$

For a cost function in the warped log spectrum domain:

$$E = \sum_{n=0}^{N-1}(\tilde{h}_n - \tilde{h}_n^*)^2 \tag{13}$$

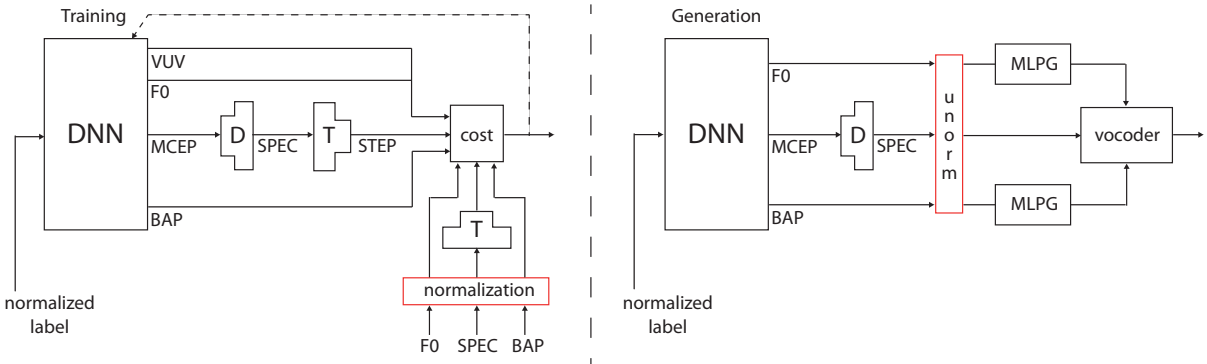

Figure 1: *Training and generation for DNN-step. D and T represent the transformation from Mel cepstral coefficients to spectrum and spectrum to STEP respectively.*

the gradient becomes:

$$\frac{\partial E}{\partial \mathbf{c}} = \frac{\partial \sum_{n=0}^{N-1} (\tilde{h}_n - \tilde{h}_n^*)^2}{\partial \mathbf{c}} \tag{14}$$

$$= \sum_{n=0}^{N-1} 2(\tilde{h}_n - \tilde{h}_n^*) \frac{\partial \tilde{h}_n}{\partial \mathbf{c}} \tag{15}$$

$$= \sum_{n=0}^{N-1} 2(\tilde{h}_n - \tilde{h}_n^*) \, \mathbf{d}_n^\top \tag{16}$$

where $\mathbf{d}_n$ is the n-th row of the matrix $\mathbf{D}_0$.

When the error is calculated in the STEP domain, the gradient with respect to the predicted Mel cepstral coefficients can be calculated as follows:

$$\frac{\partial E}{\partial \mathbf{c}} = \sum_{f=0}^{F-1} 2(y_f - y_f^*) \frac{\partial y_f}{\partial \mathbf{c}} \tag{17}$$

$$\frac{\partial y_f}{\partial \mathbf{c}} = \frac{\partial \mathbf{h}}{\partial \mathbf{c}} \frac{\partial y_f}{\partial \mathbf{h}} = \mathbf{D}_\alpha^\top \frac{\partial y_f}{\partial \mathbf{h}} = \mathbf{D}_\alpha^\top \frac{\partial \overline{\mathbf{h}}}{\partial \mathbf{h}} \frac{\partial y_f}{\partial \overline{\mathbf{h}}} \tag{18}$$

$$\frac{\partial y_f}{\partial \overline{\mathbf{h}}} = \frac{2\, s_f}{N^2 \exp y_f} \mathbf{G}_f \mathbf{F} \, \mathrm{diag}((\mathbf{G}_f \overline{\mathbf{h}})^\top \mathbf{F}) \, \mathbf{F} \, \mathbf{b} \tag{19}$$

For a similar derivation see [24].

4. Evaluation

We now explain how we trained a variety of DNNs using these cost functions, then present objective and subjective results.

4.1. Methods

We used 2542 sentences from a British male speaker: 2400 utterances were used for training, 70 utterances for development and 72 utterances for evaluation. The waveforms are sampled at 48 kHz. We extracted the following parameters: 2049 dimension warped log spectrum extracted using STRAIGHT [25], 60 Mel cepstral coefficients (MCEP) extracted from the linear form of this spectrum, 55 dimension STEP parameters, Mel scale F_0, and 25 aperiodicity energy bands (BAP) extracted using STRAIGHT. For the frequency warping applied to spectrum and cepstral coefficients, we used $\alpha = 0.77$ to approximate the Mel scale, at this particular sampling frequency.

The DNN architecture was the same as that used in [18]: six layers of 1024 tangent hidden units and a linear output layer. The cost function was mean squared error. The initial learning rate used for each method was chosen empirically. 25 epochs of training were performed with early stopping. After 10 epochs, the learning rate was then halved at each epoch; the momentum parameter was set to 0.9. Our implementation uses Theano version 0.6 [26] and training was conducted on a GPU.

We trained single DNNs that predict band aperiodicity, fundamental frequency and Mel cepstral coefficients all together, and whose spectral cost function is defined in either the Mel cepstral (DNN-mcep), warped log spectrum (DNN-spec) or STEP domains (DNN-step). The total cost function is the summation of the sum-squared error for the non-spectral features, plus the spectral cost function.

Fig. 1 shows diagrammatically how to train and generate from DNNs, when taking the spectral cost function in the STEP domain (DNN-step), where $\mathbf{D}$ refers to the matrix multiplication that converts cepstral coefficients to spectrum and $\mathbf{T}$ refers to the transformation from spectrum to STEP. The DNN predicts all acoustic features; the transformation is only applied to the spectral acoustic features. At generation time, Mel cepstral coefficients are transformed to spectrum, unnormalised and then passed to the vocoder. To train and generate when the cost function is in the spectrum domain, a similar procedure is performed but the T block is absent. First and second order delta coefficients are also predicted for F0 and BAP, but only statics are predicted for MCEP. At generation time, the maximum likelihood parameter generation (MLPG) algorithm, using precomputed variances from the training data, is applied to generate F0 and BAP trajectories while spectrum trajectories are constructed directly from the generated acoustic features. This was done because it is not obvious how to transform delta coefficients in the MCEP domain into deltas in the STEP domain. Postfiltering in the Mel cepstral domain [9] was applied when generating the waveform.

4.2. Objective distortion measures

To compare training procedures we trained a variety of models and calculated distortion measures using the test data. Table 1 shows these objective measures in different domains together with the training configuration for each model. Results for DNN-mcep vary from those presented in [18] because the deltas of spectral features are not being predicted here. The DNN-step* system is obtained by initialising the weights of the model with the previously-trained DNN-spec network.

We can see that DNN-spec produces lower objective error in the spectrum domain when compared to DNN-mcep. We can also see that when the training algorithm is designed to minimise error in the warped log spectrum domain (DNN-spec) the error measured in the STEP domain decreases as well. In fact, a smaller STEP error was obtained when minimising spectrum error than when training to minimise STEP error (DNN-step).

Table 1: *Objective distortion measures calculated for the test sentences and overall error for sentences from the development set.*

	model parameters			distortion measures					
	epochs	learning rate	momentum	SPEC (dB)	STEP (dB)	BAP (dB)	F_0 (Hz)	V/UV (%)	overall validation error (V/UV+STEP+F_0+BAP)
DNN-mcep	25	$0.2\ 10^{-3}$	0.3	7.02	4.91	1.98	18.15	4.28	n/a
DNN-spec	25	$0.1\ 10^{-5}$	0.3	6.77	4.65	2.05	18.36	4.56	72.47
DNN-step	25	$0.2\ 10^{-4}$	0.3	8.35	5.00	1.97	10.46	4.02	70.07
DNN-step*	1	$0.1\ 10^{-5}$	0.9	7.03	4.79	2.04	11.24	4.34	72.40
DNN-step*	5	$0.1\ 10^{-5}$	0.9	7.17	4.83	2.01	11.11	4.20	71.82
DNN-step*	15	$0.1\ 10^{-5}$	0.9	7.75	4.84	1.99	10.71	4.09	71.17

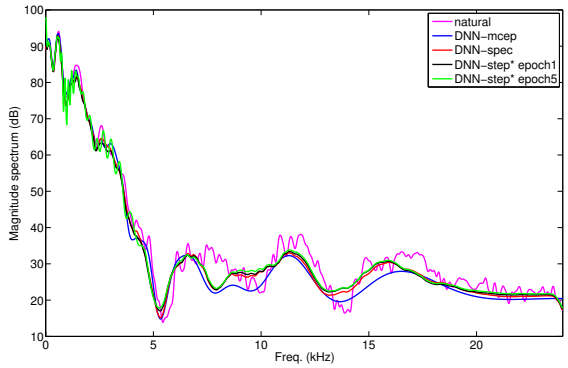

Figure 2: *Spectrum averaged across frames of a vowel segment.*

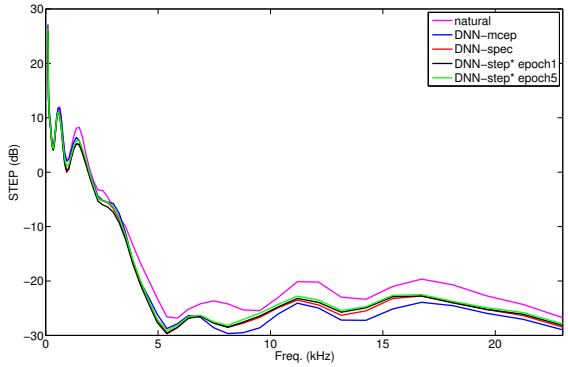

Figure 3: *STEP averaged across frames of a vowel segment.*

The overall validation error was however smaller for the DNN-step system, possibly because F0 and voiced/unvoiced (V/UV) errors were smaller. When initialising the DNN-step network with the weights and biases of the DNN-spec network (DNN-step*), the STEP error increases and F0 and V/UV errors decrease with more epochs. Note that overall validation error while training decreases with epochs, indicating convergence. Even though the STEP distortion is lower than the one obtained by the DNN-mcep model, the spectrum distortion obtained at epochs 5 and 15 while training DNN-step* is larger. We can report that speech generated from these models contained audible artefacts. Although not reported here, a similar trend was also observed for error measured using the original formulation [21] of STEP, calculated from the synthetic waveform .

Figs. 2 and 3 present the spectrum and STEP averaged across the frames of a vowel. We can see that even though DNN-step* at epoch 5 presents ripples in the spectrum domain, its STEP representation is quite close to the DNN-spec one.

4.3. Subjective results

We evaluated the systems DNN-mcep, DNN-spec and DNN-step* (epoch 1). 28 native English speakers performed a preference test rating each of the three possible pair comparison 24 times. A different sentence was used for each of the 72 pairs (within each pair, the same text was used for both synthetic utterances). The order of the sentences was made random and the pair comparison was also randomised such that at every six sentences all comparisons were covered. 24 sentences from the development set were used in a training session prior to the test.

Fig. 4 shows, for each comparison, the average preference score in % with 95% confidence intervals calculated using a two-tailed binomial test. We can see that both DNN-spec and DNN-step are preferred over DNN-mcep, with the preference for DNN-spec being significant. The DNN-spec is mildly preferred over DNN-step, although the difference is not significant.

5. Conclusions

We propose a general formulation for minimum perceptual error training of DNNs for speech synthesis. We trained DNNs that predict vocoder parameters and used spectral cost functions in the Mel cepstral, warped log spectrum or a perceptually-oriented domain. We note that our framework could be used with to a variety of other perceptually-oriented domains not just STEP.

Calculating the cost in the spectrum domain generates speech that is most preferred by listeners. The question of which domain is best for measuring perceptual differences remains open. Looking across the field of speech processing, we see that appropriate use of expert knowledge – such as cochlea frequency resolution – has led to great improvements. We believe that, if we can find the correct way, knowledge about hair cell sensitivity, frequency masking and temporal masking will also lead to improvements. Here, we tried one particular model that accounts for some of these aspects, but that did not yet provide improvements over the spectrum. Future work includes considering different domains to represent speech.

Acknowledgements This work was supported by EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

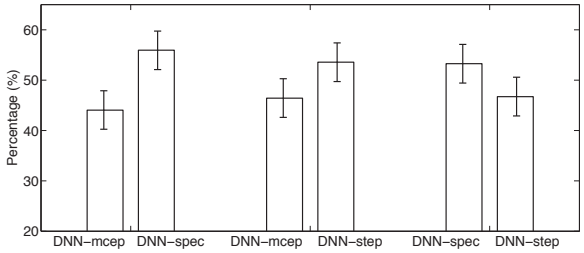

Figure 4: *Preference test results.*

6. References

[1] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Comp. Speech and Lang.*, vol. 21, no. 1, pp. 153–173, 2007.

[2] M. Shannon, H. Zen, and W. Byrne, "Autoregressive models for statistical parametric speech synthesis," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 21, no. 3, pp. 587–597, March 2013.

[3] Y.-J. Wu and R.-H. Wang, "Minimum generation error training for HMM-Based speech synthesis," in *Proc. ICASSP*, Toulouse, France, May 2006, pp. I89–I92.

[4] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 5, pp. 816–824, 2007.

[5] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, Vancouver, Canada, 2013, pp. 7962–7966.

[6] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proc. ICASSP*, Florance, Italy, 2014, pp. 3872–3876.

[7] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio, and P. Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 153–165, Jan. 2011.

[8] D. Erro, I. Sainz, E. Navas, and I. Hernaez, "Harmonics plus noise model based vocoder for statistical parametric speech synthesis," *J. Sel. Topics in Sig. Proc.*, vol. 8, no. 2, pp. 184–194, April 2014.

[9] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Incorporating a mixed excitation model and postfilter into HMM-based text-to-speech synthesis," *Systems and Computers in Japan*, vol. 36, no. 12, pp. 43–50, 2005.

[10] S. Takamichi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "A postfilter to modify the modulation spectrum in HMM-based speech synthesis," in *Proc. ICASSP*, May. 2014.

[11] L.-H. Chen, T. Raitio, C. Valentini-Botinhao, J. Yamagishi, and Z.-H. Ling, "DNN-based stochastic postfilter for HMM-based speech synthesis," in *Proc. Interspeech*, 2014, pp. 1954–1958.

[12] G. E. Henter, T. Merritt, M. Shannon, C. Mayo, and S. King, "Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech," in *Proc. Interspeech*, vol. 15, September 2014, pp. 1504–1508.

[13] Y.-J. Wu and K. Tokuda, "Minimum generation error training by using original spectrum as reference for log spectral distortion measure," in *Proc. ICASSP*, Taipei, Taiwan, April 2009, pp. 4013–4016.

[14] K. Nakamura, K. Hashimoto, Y. Nankaku, and K. Tokuda, "Integration of acoustic modeling and mel-cepstral analysis for hmm-based speech synthesis," in *Proc. ICASSP*, May 2013, pp. 7883–7887.

[15] S. Takaki and J. Yamagishi, "Constructing a deep neural network based spectral model for statistical speech synthesis," in *NOLISP (submitted)*, 2015.

[16] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. ICASSP*, vol. 1, San Francisco, USA, March 1992, pp. 137–140.

[17] Y. Qian, Y. Fan, W. Hu, and F. Soong, "On the training aspects of Deep Neural Network (DNN) for parametric TTS synthesis," in *Proc. ICASSP*, May 2014, pp. 3829–3833.

[18] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proc. ICASSP*, Brisbane, Australia, 2015.

[19] S. Imai and C. Furuichi, "Unbiased estimator of log spectrum and its application to speech signal processing," in *Proc. EURASIP*, Grenoble, France, Sep. 1988, pp. 203–206.

[20] M. Cooke, "Glimpsing speech," *Journal of Phonetics*, vol. 31, pp. 579 – 584, 2003.

[21] ——, "A glimpsing model of speech perception in noise," *J. Acoust. Soc. Am.*, vol. 119, no. 3, pp. 1562–1573, 2006.

[22] C. Valentini-Botinhao, J. Yamagishi, and S. King, "Can objective measures predict the intelligibility of modified HMM-based synthetic speech in noise?" in *Proc. Interspeech*, Florence, Italy, August 2011, pp. 1837 – 1840.

[23] B. C. J. Moore and B. R. Glasberg, "A revision of Zwicker's loudness model," *Acta Acustica*, vol. 82, pp. 335–345, 1996.

[24] C. Valentini-Botinhao, J. Yamagishi, S. King, and R. Maia, "Intelligibility enhancement of HMM-generated speech in additive noise by modifying mel cepstral coefficients to increase the glimpse proportion," *Comp. Speech and Lang.*, vol. 28, no. 2, pp. 665–686, 2014.

[25] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds," *Speech Comm.*, vol. 27, pp. 187–207, 1999.

[26] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proc. of the Python for Scientific Computing*, Jun. 2010.